Visual Comparator

An Interactive Tool for Dynamic Spatiotemporal Comparative Visualizations

Nihanth W. Cherukuru and Tim Scheitlin

ABSTRACT: Visualizations enable us to detect patterns, time-evolving features, and trends in complex datasets that might not be obvious by looking at the raw data. The visual exploration process often requires comparisons between multiple visualizations, either from the same dataset or a different one, to identify relationships and patterns. This visualization process, referred to as comparative visualization, is valuable for analyzing multivariate, multispectral, or multidimensional data. The existing tools that facilitate visual comparisons do this by three means: juxtaposition (placing visuals side by side), superposition (overlaying visuals), and explicit encoding (visualizing a derived quantity corresponding to the relationship being studied). While superposition is ideal for static, geospatial datasets, where spatialization is a key component of the data, the spatio-temporal nature of Earth science datasets presents a challenge with comparative visualizations. Visual Comparator is an interactive, cross-platform (desktops, kiosks, and web), open-source application, developed to address this shortcoming. The application is used to superimpose and compare up to three synchronized, animated visualizations, and a transition between the visualizations is provided through a slider-based interface. This form of visualization has the advantage of drawing the viewers' attention to changes between the datasets, enabling comparisons of scale, and reducing the clutter caused by having multiple variables in one visual. This article provides an overview of the project, a brief review of literature pertaining to human perception research and comparative visualizations, and a guide to accessing this application.

<https://doi.org/10.1175/BAMS-D-19-0266.1>

Corresponding author: Nihanth W. Cherukuru, ncheruku@ucar.edu

In final form 19 May 2020

©2020 American Meteorological Society

For information regarding reuse of this content and general copyright information, consult the [AMS Copyright Policy](#).

AFFILIATIONS: Cherukuru and Scheitlin—National Center for Atmospheric Research, Boulder, Colorado

The advancements in measurement techniques and computational capabilities have made high-resolution, spatiotemporal, multivariate data ubiquitous. While novel techniques, such as those involving artificial intelligence and machine learning have shown promising results in big data analytics, there is always a need for tools that facilitate visual analysis that complement the computational approach. Technological advancements have enabled us to create visualization tools that tap into intuitive human perception, enabling us to detect patterns and trends in complex datasets that might not be obvious by looking at the raw data alone (Cherukuru and Calhoun 2016; Cherukuru et al. 2017). The visual exploration process to identify relationships and detect patterns, often requires the users to work with multiple visualizations necessitating comparative visualization tools. The general idea of comparative visualization refers to the process of visually depicting the similarities or differences, either implicitly or explicitly, between multiple data sources (Pagendarm and Post 1995). Comparative visualization techniques have been proven to be valuable in myriad application domains such as genetic sequencing, flow visualization, medical imaging, GIS, network analysis, and image processing (Gleicher 2018). Despite these developments, most of the solutions address domain-specific requirements, limiting their broader application.

Data pertaining to atmospheric and related sciences are increasingly multivariate, multispectral, or multidimensional in nature. While comparative visualization systems would be beneficial for geospatial datasets, where spatialization is a key component of the data, the spatiotemporal nature of Earth science datasets presents a challenge with comparative visualizations. To the best of our knowledge, there are no general-purpose tools that facilitate comparative visual analysis for spatiotemporal data. The Visual Comparator was developed to address this shortcoming. It is an open-source, postprocessing viewer to superimpose multiple animations and interactively reveal selected portions of each visualization via a slider. It was designed to be a user-friendly, cross-platform application that could be used by domain specialists as well as a broad range of users who are interested in comparing animated visualizations. This article is organized into two sections. The first gives an overview of literature pertaining to human perception and available comparative visualization systems along with the rationale for developing this tool, followed by a high-level description of Visual Comparator, and a link to software available for researchers, developers, and interested users.

Relevant literature

Limitations in human visual perception and memory make the comparative visualization process a difficult task (Scott-Brown et al. 2000). Our inability to detect even surprisingly large changes in the visual stimuli has been reported in the previous studies, thereby requiring systems that support and specifically address the comparison tasks (Franconeri 2013). There are two approaches to the general idea of comparative visualization: image-level comparison, and data-level comparison (Pagendarm and Post 1995). In image-level comparison, visualizations/images are generated from the source through separate visualization pipelines and the resulting images are then used for comparison. In data-level comparison, the data from different sources are fed into a common visualization pipeline that generates the combined visuals for comparisons. The former has the advantage of accommodating dissimilar datasets (such as observational data and simulation data) and is beneficial for exploratory analysis where the relationship between the variables is not known a priori. Gleicher et al. (2011) proposed a general taxonomy of visual designs for explicitly assisting with comparison, based on a survey of existing tools across multiple application domains and data types. The designs were grouped into three categories: juxtaposition, superposition, and explicit encoding. Examples of each are shown in Fig. 1.

Juxtaposition. Juxtaposition designs facilitate comparative exploration by placing the visuals side by side (Fig. 1a). This design relies on the viewer's memory to hold information from multiple windows and make connections. Also known as small multiples (Tufte et al. 1990) or multiple views layout (Baldonado et al. 2000), these designs are popular owing to their simplicity and being one of the best compromises for representing animations in static print medium. However, the limitation of screen space and the reliance on mental integration for comparisons increases the demand on cognitive attention (Baldonado et al. 2000). Juxtaposition designs are more suitable in instances where the data being displayed involve sufficient context switching between the views; i.e., it is easier to identify the differences through juxtaposition when the visualizations are very dissimilar to one another (Ryu et al. 2003). This limits the benefits to situations where the comparisons take place *within the eye span* (Tufte et al. 1990).

Fig. 1. The three types of comparative visualization designs (Gleicher et al. 2011). (a) Juxtaposition as applied to (a1) graphs and (a2) maps, in which the graphics are placed side by side in individual panels for comparison. (b) Superposition as applied to (b1) graphs and (b2) maps. In these designs, the graphics are spatially overlaid on one another. The map in (b2) uses variations in opacity to show the terminator line and the boundary layer height (colored). (c) Explicit encoding as applied to (c1) graphs and (c2) maps. These designs visually encode the relationship under study. For instance, the difference of the two plots from the previous subplots are shown in (c1) (data 2 minus data 1) and (c2) shows the accumulated precipitation (blue), which is the time integral of the quantity overlaid in red. It must be noted that all cartographic visualizations inherently use superposition since the data are overlaid on a map; thus, (a2), (b2), and (c2) have features of superposition designs.

Superposition. Superposition designs employ techniques such as overlays, to display multiple datasets within the same window or viewing space (Fig. 1b). This design is suitable in instances where different data sources reside in the same space and where spatialization is an important attribute of the data, such as those commonly encountered in geospatial datasets and maps. Also known as visual multiplexing (Chen et al. 2014), these designs rely on our improved ability to detect patterns between visuals requiring minimal eye movement and memory load. This phenomenon was observed in studies by Muehrcke et al. (1978) and Dill and Fahle (1998), which showed that patterns are more easily detected when the visual stimuli are presented at the same, rather than at different, locations. Although superposition designs most commonly employ overlays and transparencies, there have been works that involve other techniques such as color weaving and attribute blocks—interweave different datasets into one visual giving the appearance of a weaved tapestry (Hagh-Shenas et al. 2007; Miller 2007); texture stitching—a technique that preserves occluded regions/map boundaries by adjusting the spatial frequency of the overlay (Urness et al. 2003); color and texture compositing—combining colors with textures to represent multiple collocated variables (Shenas and Interrante 2005); and icons—a map with icons whose color, shape, size, orientation, pattern, and texture are used to encode different fields (Zhang and Pazner 2004). Despite the perceptual advantages of superposition designs, issues with clutter and occlusion caused by multiple visuals limit their use to a maximum of two to three datasets (Gleicher 2018). This is particularly pronounced with continuous variables and data with high density. Tominski et al. (2012) designed a system to address the issue with occlusion by designing an interface based on the real-world behavior of people comparing information on printed paper for comparisons. While techniques like color weaving, texture compositing, and iconography were meant to address this issue, they are useful only if the data have large uniformly valued regions (Shenas and Interrante 2005), which is a limitation when visualizing continuous fields.

Explicit encoding. Designs employing explicit encoding, visually encode (compute) the underlying relationship between the visuals and use that as a basis for comparisons. Unlike juxtaposition and superposition, which rely on the viewer, explicit encoding uses computation to facilitate comparisons (Fig. 1c). While this design offers a straightforward solution to the viewer to perform the task at hand, it requires the relationship to be known a priori, which might not be ideal for exploratory analysis. Moreover, these designs often require a mechanism to relate the visual, back to the underlying data, or could suffer from decontextualization. Consequently, pure explicit encoding designs are seldom used alone and are often presented with a visual of the underlying data through juxtaposition or superposition (Gleicher et al. 2011). An example of this type of design is the Visual Analysis for Image Comparison (VAICo) system (Schmidt et al. 2013), which is an interactive tool to visualize image variances.

There have been a number of hybrid designs, which incorporate a combination of juxtaposition, superposition, and explicit encoding, as a means to address the limitations of individual designs. A user evaluation study conducted by Srinivasan et al. (2018) found that designs involving combinations of the methodologies perform the same or better than individual designs. Most of the preceding examples dealt with static data/images facilitating comparison in a spatial context. Attempts have been made to support visual comparison in a temporal context. This functionality was provided often through juxtaposed synchronized animations. However, Blok et al. (1999) investigated design options for visual exploration of cartographic spatiotemporal data and identified the difficulty of juxtaposition designs for exploration in this context. It is convenient to compare heterogeneous behaviors in animated data, using overlapping displays (superposition) as opposed to juxtaposition (Andrienko et al. 2003).

While superposition is ideal for geospatial datasets, where spatialization is a key component of the data, the spatiotemporal nature of Earth science datasets presents a challenge with comparative visualization and to the best of our knowledge, there are not any tools that provide this functionality in a user-friendly, general purpose application. The Visual Comparator was developed to address this shortcoming. The closest functionality to the tool presented in this article, is provided in the Satellite Loop Interactive Data Explorer in Real-Time (SLIDER) application (Micke 2018). SLIDER is an interactive web-based tool with an option for overlaying and comparing satellite data by varying the opacity of layers along with a slider. The application presented in this article differs significantly from the previous study in a couple of aspects. While SLIDER is designed specifically for visualizing high-resolution satellite data, the Visual Comparator is a generic postprocessing tool that could be incorporated in multiple application domains.

Description of the tool

The Visual Comparator is a postprocessing tool; i.e., the application works with video files/animations generated from any visualization software (Fig. 2). Support for image sequences would be added in the near future. The application synchronizes the video streams and superimposes up to three animations, allowing comparison with an interactive, slider-based interface that enables the user to reveal/hide portions of each animation. The direction of the slider is interchangeable (horizontal or vertical). A screenshot of the application's user interface (UI) is shown in Fig. 3. To synchronize the input files, an external clock is used to set the playback speed of all the animations and textures are generated for each corresponding frame from the input animations. These textures are overlaid on one another and revealed by proportional scaling controlled by a slider. Playback controls are handled by controlling the external clock (Fig. 2). The application assumes that all animations have the same duration and total frame count. This is enforced through a duration and total frame count check prior to playback. While this is a valid assumption for most of the situations, it could be a limitation while working with historic animations and data originating from different sources recorded with different frame rates. One interim solution for this limitation would be to use a video editing software to modify the frame rate of the animation before using it in the Visual Comparator. The application is compatible with most of the commonly used video formats such as *.asf, *.avi, *.dv, *.m4v, *.mov, *.mp4, *.mpg, *.mpeg, *.ogv, *.vp8, *.webm, and *.wmv, provided they are supported

Fig. 2. Overview of the application. The animations are synchronized to an external clock and textures are generated for each corresponding frame from the input animations. These textures are overlaid on one another and revealed by proportional scaling controlled by a slider. Playback controls are handled by controlling the external clock.

Fig. 3. Visual Comparator user interface. A screenshot of the application displaying precipitable water vapor overlaid on sea surface temperature (SST). The user can reveal the underlying SST values using the slider.

by the target platform. The application is currently available as a desktop (PC and Mac) and a web application. While the user interface is identical in both the versions, they differ in the data selection and initialization process. The desktop version allows the user to interactively select the animations to be included in the viewer, whereas the web version is initialized from a JavaScript Object Notation (JSON) file (see Fig. 4). The web application can be embedded in web pages and is compatible with any browser that supports WebGL content such as Chrome, Firefox, and Safari.

The project was developed using Unity (Unity Technologies 2020).

Archived examples of use cases (Fig. 5) can be accessed from the GitHub project page. Some of the use cases include temporal comparisons (e.g., El Niño in 1997 and 2013), spatial comparisons (e.g., observed Arctic and Antarctic sea

Fig. 4. Visual Comparator requires the path to the animation files, the direction of the slider, and an optional short description (tag) for each animation that is ingested into the application. These inputs can be entered through (a) an interactive window for the desktop application and (b) a startup JSON file included in the application's streaming assets directory for the web version. A screenshot of the desktop application with the file selection window is shown in (a), and (b) shows a sample JSON code snippet that is configured to run two animations.

Fig. 5. Example use cases. (a) Spatial comparison of data displaying SST and sea ice extent in the Northern Hemisphere (top half of panel) and Southern Hemisphere (bottom half of panel). (b) Observed (top half of panel) and modeled (bottom half of panel) SST and sea ice extent in the Southern Hemisphere. (c) Temporal comparison of the El Niño event from 1997 (left half of panel) and 2015 (right half of panel). (d) The Atmospheric Imaging Assembly (AIA) images of the sun in multiple wavelengths: 304 Å (top third of panel), 193 Å (middle third of panel), and 171 Å (bottom third of panel). Courtesy of NASA Solar Dynamics Observatory (SDO) and the AIA, Extreme Ultraviolet Variability Experiment (EVE), and Helioseismic and Magnetic Imager (HMI) science teams. The examples shown here can be accessed from the Visual Comparator project page (see the “How to access the examples, source code, and binaries” sidebar).

ice change), multivariate datasets (e.g., multispectral images of the sun), and studying the effect of resolution on datasets.

Summary

The Visual Comparator was received enthusiastically by both scientific and the education and outreach communities, receiving positive feedback. It is important to emphasize that the Visual Comparator was designed not as a data visualization tool but rather a viewer with the sole purpose of facilitating slider-based comparisons. As a postprocessing tool, this application

How to access the examples, source code, and binaries

Visual Comparator is an open-source project developed using the Unity game engine (Unity Technologies 2020).

Project home page (case sensitive)

<https://ncar.github.io/VisualComparator/>

Compatibility

Desktop application: PC, Mac

Web application: Safari, Chrome, and Firefox (only on laptops and desktops)

Kiosk: PC

Supported file formats

*.asf, *.avi, *.dv, *.m4v, *.mov, *.mp4, *.mpg, *.mpeg, *.ogv, *.vp8, *.webm, and *.wmv (platform-specific limitations apply)

allows users to adhere to their familiar visualization pipeline to generate the image sequences/animations that can then be ingested into the Visual Comparator. Consequently, the application is agnostic of the raw data and numerous domain-specific data formats, allowing for simple controls and easy to use interface making it accessible to a wider community besides domain experts. This implementation also has an added advantage of finding potential applications with older visualizations where the original visualization project/software is not readily accessible. Many modern devices have inbuilt hardware acceleration and platform-specific optimizations for encoding/decoding videos. The use of game engines such as Unity (Unity Technologies 2020) for application development enables us to utilize these features without having to directly work with platform-specific, native custom application programming interfaces (APIs), making the development and maintenance more efficient. This is an open-source project and the source code and binaries are available for use to other researchers and interested users (see the “How to access the examples, source code, and binaries” sidebar to download and/or contribute to the project).

Acknowledgments. The authors thank Matt Rehme for lending the visualizations used in the example files and testing the application. This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement 1852977.

References

- Andrienko, N., G. Andrienko, and P. Gatalsky, 2003: Exploratory spatio-temporal visualization: An analytical review. *J. Visual Lang. Comput.*, **14**, 503–541, [https://doi.org/10.1016/S1045-926X\(03\)00046-6](https://doi.org/10.1016/S1045-926X(03)00046-6).
- Baldonado, M. Q. W., A. Woodruff, and A. Kuchinsky, 2000: Guidelines for using multiple views in information visualization. *Proc. Working Conf. on Advanced Visual Interfaces*, Palermo, Italy, Association for Computing Machinery, 110–119, <https://doi.org/10.1145/345513.345271>.
- Blok, C., B. Köbben, T. Cheng, and A. A. Kuterema, 1999: Visualization of relationships between spatial patterns in time by cartographic animation. *Cartogr. Geogr. Inf. Sci.*, **26**, 139–151, <https://doi.org/10.1559/152304099782330716>.
- Chen, M., S. Walton, K. Berger, J. Thiyaalingam, B. Duffy, H. Fang, C. Holloway, and A. E. Trefethen, 2014: Visual multiplexing. *Comput. Graph. Forum*, **33**, 241–250, <https://doi.org/10.1111/cgf.12380>.
- Cherukuru, N. W., and R. Calhoun, 2016: Augmented reality based Doppler lidar data visualization: Promises and challenges. *EPJ Web Conf.*, **119**, 14006, <https://doi.org/10.1051/epjconf/201611914006>.
- , T. Scheitlin, M. Rehme, and R. R. P. Kumar, 2017: Atmospheric data visualization in mixed reality. *Bull. Amer. Meteor. Soc.*, **98**, 1585–1592, <https://doi.org/10.1175/BAMS-D-15-00259.1>.
- Dill, M., and M. Fahle, 1998: Limited translation invariance of human visual pattern recognition. *Percept. Psychophys.*, **60**, 65–81, <https://doi.org/10.3758/BF03211918>.
- Franconeri, S. L., 2013: The nature and status of visual resources. *Oxford Handbook of Cognitive Psychology*, Vol. 8481, Oxford University Press, 147–162, <https://doi.org/10.1093/oxfordhb/9780195376746.013.0010>.
- Gleicher, M., 2018: Considerations for visualizing comparison. *IEEE Trans. Visualization Comput. Graph.*, **24**, 413–423, <https://doi.org/10.1109/TVCG.2017.2744199>.
- , D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, 2011: Visual comparison for information visualization. *Inf. Visualization*, **10**, 289–309, <https://doi.org/10.1177/1473871611416549>.
- Hagh-Shenas, H., S. Kim, V. Interrante, and C. Healey, 2007: Weaving versus blending: A quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. *IEEE Trans. Visualization Comput. Graph.*, **13**, 1270–1277, <https://doi.org/10.1109/TVCG.2007.70623>.
- Micke, K., 2018: Every pixel of GOES-17 imagery at your fingertips. *Bull. Amer. Meteor. Soc.*, **99**, 2217–2219, <https://doi.org/10.1175/BAMS-D-17-0272.1>.
- Miller, J. R., 2007: Attribute blocks: Visualizing multiple continuously defined attributes. *IEEE Comput. Graph. Appl.*, **27**, 57–69, <https://doi.org/10.1109/MCG.2007.54>.
- Muehrcke, P. C., J. O. Muehrcke, and A. J. Kimerling, 1978: *Map Use: Reading, Analysis, and Interpretation*. Esri Press, 469 pp.
- Pagendarm, H. G., and F. H. Post, 1995: *Comparative Visualization—Approaches and Examples*. Delft University of Technology, 28 pp.
- Ryu, Y. S., B. Yost, G. Convertino, J. Chen, and C. North, 2003: Exploring cognitive strategies for integrating multiple-view visualizations. *Proc. Human Factors and Ergonomics Society Annual Meeting*, Los Angeles, CA, Human Factors and Ergonomics Society, 591–595, <https://doi.org/10.1177/154193120304700371>.
- Schmidt, J., M. E. Gröller, and S. Bruckner, 2013: VAICo: Visual Analysis for Image Comparison. *IEEE Trans. Visualization Comput. Graph.*, **19**, 2090–2099, <https://doi.org/10.1109/TVCG.2013.213>.
- Scott-Brown, K. C., M. R. Baker, and H. S. Orbach, 2000: Comparison blindness. *Visual Cognit.*, **7**, 253–267, <https://doi.org/10.1080/135062800394793>.
- Shenas, H. H., and V. Interrante, 2005: Compositing color with texture for multi-variate visualization. *Proc. Third Int. Conf. on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, Dunedin, New Zealand, Association for Computing Machinery, 443–446, <https://doi.org/10.1145/1101389.1101478>.
- Srinivasan, A., M. Brehmer, B. Lee, and S. M. Drucker, 2018: What's the difference?: Evaluating variations of multi-series bar charts for visual comparison tasks. *Proc. 2018 CHI Conf. on Human Factors in Computing Systems*, Montreal, QC, Canada, Association for Computing Machinery, 304, <https://doi.org/10.1145/3173574.3173878>.
- Tominski, C., C. Forsell, and J. Johansson, 2012: Interaction support for visual comparison inspired by natural behavior. *IEEE Trans. Visualization Comput. Graph.*, **18**, 2719–2728, <https://doi.org/10.1109/TVCG.2012.237>.
- Tufte, E. R., N. H. Goeler, and R. Benson, 1990: *Envisioning Information*. Graphics Press, 126 pp.
- Unity Technologies, 2020: Unity core platform. Accessed 18 May 2020, <https://unity.com/products/core-platform>.
- Urness, T., V. Interrante, I. Marusic, E. Longmire, and B. Ganapathisubramani, 2003: Effectively visualizing multi-valued flow data using color and texture. *Proc. 14th IEEE Visualization 2003*, Seattle, WA, IEEE, 115–121, <https://doi.org/10.1109/VISUAL.2003.1250362>.
- Zhang, X., and M. Pazner, 2004: The icon imagemap technique for multivariate geospatial data visualization: Approach and software system. *Cartogr. Geogr. Inf. Sci.*, **31**, 29–41, <https://doi.org/10.1559/152304004773112758>.